LEGIBILITY NOTICE

LA-UR-87-1863

LA-UR--87-1863

DE87 011793

CONF-870775--1

TITLE GENETIC ALGORITHMS AND CLASSIFIER SYSTEMS: FOUNDATIONS AND FUTURE DIRECTIONS

AUTHOR(S) John H. Holland

SUBMITTED TO Proceedings of the "International Conference on Genetic Algorithms and Their Applications" to be held at M.I.T., Cambridge, Massachusetts, July 27-29, 1987.

Genetic Algorithms and Classifier Systems: Foundations and Future Directions

John H. Holland

The University of Michigan

Abstract

Theoretical questions about classifier systems, with rare exceptions, apply equally to other *adaptive nonlinear networks* (ANNs) such as the connectionist models of cognitive psychology, the immune system, economic systems, ecologies, and genetic systems. This paper discusses pervasive properties of ANNs and the kinds of mathematics relevant to questions about these properties. It discusses relevant functional extensions of the basic classifier system and extensions of the extant mathematical theory. An appendix briefly reviews some of the key theorems about classifier systems.

Classifier systems are examples of a broad class of systems sometimes called *adaptive nonlinear networks* (ANNs, hereafter). Broadly characterized, classifier systems, and ANNs in general, consist of a large number of units that (1) interact in a nonlinear, competitive fashion, and (2) are modified by various operators so that the system as whole progressively adapts to its environment. Typically an ANN confronts an environment that exhibits perpetual novelty and it can function (or continue to exist) only by making continued adaptations to that environment. Because ANN/environment interactions are complex, except in artificially constrained cases, an ANN usually operates far from equilibrium. ANNs form the core of areas of study as diverse as cognitive psychology, artificial intelligence, economics, immunogenesis, genetics, and ecology.

Foundations.

Classifier systems are quite typical ANNs so that questions about classifiers, suitably translated, are typically questions about

ANNs and vice versa. To carry out the translation it is necessary to identify the counterparts in other ANNs of the message-processing rules, called *classifiers*, that are the units of classifier is necessary to identify the counterparts in other ANNs of the message-processing rules, called *classifiers*, that are the units of classifier systems. For example: In genetics, the counterparts are chromosomes; in game theory and economics, they are (rule-defined) strategies; in immunogenesis, antigens; in connectionist versions of cognition, (formally-defined) neurons; and so on. Under such translation it is relatively easy to identify a range of important theoretical questions that apply to classifier systems in particular and ANNs in general:

[These questions, and some of the ensuing discussions, are presented assuming that the reader has some familiarity with Holland et al. [1986] or Holland [1975]. There is simply not enough room here to define the terms; a reader familiar with the literature concerning some other ANN should be able to make the relevant translation in most cases.]

(1) What parameters and operators favor the emergence of stable hierarchical covers such as default hierarchies, internal models, and the like (via an increased diversity of units and progressively more complicated interactions between them)?

(2) Are the familiar "ecological" interactions -- parasitism, symbiosis, competitive exclusion, etc. -- a common feature of all parallel, nonlinear competitive systems?

(3) Are multi-functional units (units that can serve in several contexts) the major stepping-stone employed by all ANNs in making adaptive advances?

(4) What environmental conditions favor recombination, imprinting, triggering, and other constrained or biased procedures for generating new trials (rules, chromosomes, organizational structures, etc.)?

(5) What environmental conditions favor tracking vs averaging, exploration vs. exploitation, etc.?

(6) What combinations of operators yield implicit parallelism?

Traditional mathematics with its reliance upon linearity, convergence, fixed points, and the like, seems to offer few tools for studying such questions. Yet, without a relevant mathematical framework, there is less chance of understanding ANNs than there would be of understanding physical phenomena in the absence of guidance from theoretical physics. A mathematics that puts emphasis on combinatorics and competition between parallel processes is the key to understanding ANNs. What seems startling when one uses differential equations, where the emphasis is on continuity, is comonplace in a programming or recursive format, where the emphasis is upon combinatorics. (Consider, for example, the chaotic regimes that are so unexpected in the context of differential equations, but are an everday occurrence, in the guise of biased random number generators, in the programming context.)

Because classifier systems are formally defined and computer-oriented, with an emphasis on combination and competition, they offer a useful test-bed for both mathematical and simulation studies of ANNs. We already have some theorems that provide a deeper understanding of the behavior of classifier systems (see the Appendix), and simulations suggest a broader class of theorems that delineate the conditions under which internal models (q-morphisms) emerge in response to complex environments (Holland [1986b]).

By putting classifier systems in a broader context, we can bring to bear relevant pieces of mathematics from other studies. For instance, in mathematical economics there are pieces of mathematics that deal with (1) hierarchical organization, (2) retained earnings (fitness) as a measure of past performance, (3) competition based on retained earnings, (4) distribution of earnings on the basis of local interactions of consumers and suppliers, (5) taxation as a control on efficiency, and (6) division of effort between production and research (exploration versus exploration). Many of these fragments, mutatis mutandis, can be used to study the counterparts of these processes in other ANNs.

As another example, in mathematical ecology there are pieces of mathematics dealing with (1) niche exploitation (models exploiting environmental opportunities), (2) phylogenetic hierarchies, polymorphism and enforced diversity (competing subsystems), (3) functional convergence (similarities of subsystem organization enforced by environmental requirements on payoff attainment), (4) symbiosis, parasitism, and mimicry (couplings and interactions in a default hierarchy, such as an increased efficiency for extant generalists simply because related specialists exclude them from some regions in which they are inefficient), (5) food chains, predator-prey relations, and other energy transfers (apportionment of energy or payoff amongst component subsystems), (6) recombination of multifunctional co-adapted sets of genes (recombination of building blocks), (7) assortative mating (biased or triggered recombination), (8) phenotypic markers affecting interspecies and intraspecies interactions (coupling), (9) "founder" effects (generalists giving rise to specialists), and (10) other detailed commonalities such as tracking versus averaging over environmental changes (compensation for environmental variability), allelochemicals (cross-inhibition), linkage (association and encoding of features), and still others. Once again, though mathematical ecology is a young science, there is much in the mathematics that has been developed that is relevant to the study of other nonlinear systems far from equilibrium.

The task of theory is to explain the pervasiveness of these features by elucidating the general mechanisms that assure their emergence and evolution. Properly applied to classifier systems, or to ANNs in general, such a theory militates against ad hoc solutions, assuring robustness and adaptability for the resulting organization. One of the best ways to insure that the mechanisms investigated are general is "to look over your shoulder" frequently to see if the mechanisms apply to all ANNs. This view is sharpened if we pay close attention to features shared by all ANNs:

(1) *Hierarchical organization*. All ANNs exhibit an hierarchical organization. In living systems proteins combine to form organelles, which combine to form cell types, and so on, through organs, organisms, species, and ultimately ecologies. Economies involve individuals, departments, divisions, companies, economic sectors, and so on, until one reaches national, regional, and world economies. A similar story can be told for each of the areas cited. These structural similarities are more than super-

ficial. A closer look shows that the hierarchies are constructed on a "building block" principle: Subsystems at each level of the hierarchy are constructed by combination of small numbers of subsystems from the next lower level. Because even a small number of building blocks can be combined in a great variety of ways there is a great space of subsystems to be tried, but the search is biased by the building blocks selected. At each level, there is a continued search for subsystems that will serve as suitable building blocks at the next level.

(2) *Competition*. A still closer look shows that in all cases the search for building blocks is carried out by competition in a population of candidates. Moreover there is a strong relation between the level in the hierarchy and the amount of time it takes for competitions to be resolved -- ecologies work on a much longer time-scale than proteins, and world economies change much more slowly than the departments in a company. More carefully, if we associate random variables with subsystem ratings (say fitnesses), then the sampling rate decreases as the level of the subsystem increases. As we will see, this has profound effects upon the way in which the system moves through the space of possibilities.

(3) *Game-like system/environment interaction*. An ANN interacts with its environment in a game-like way: Sequences of action ("moves") occasionally produce *payoff*, special inputs that provide the system with the wherewithall for continued existence and adaptation. Usually payoff can be treated as a simple quantity (energy in physics, fitness in genetics, money in economics, winnings in game theory, reward in psychology, error in control theory, etc.) It is typical that payoff is sparsely distributed in the environment and that the adaptive system must compete for it with other systems in the environment.

(4) *Exploitation of regularities*. The environment typically exhibits a range of regularities or *niches* that can be exploited by different action sequences or *strategies*. As a result the environment supports a variety of processes that interact in complex ways, much as in a multi-person game. Usually there is no super-process that can outcompete all others so an ecology results (domains in physics, interacting species in ecological genetics, companies in economics, cell assemblies in neurophysiological psychology, etc.). The very complexity of

these interactions assures that even large systems over long time spans can have explored only a minuscule range of possibilities. Even for much-studied board games such as chess and go this is true; the not so simply defined "games" of ecological genetics, economic competition, immunogenesis, CNS activity, etc., are orders of magnitude more complex. As a consequence the systems are always far from any optimum or equilibrium situation.

(5) *Exploration vs. exploitation.* There is a tradeoff between *exploration* and *exploitation.* In order to explore a new niche a system must use new and untried action sequences that take it into new parts (state sets) of the enviroment. This can only occur at the cost of departing from action sequences that have well-established payoff rates. The ratio of exploration to exploitation in relation to the opportunities (niches) offered by the environment has much to do with the life history of a system.

(6) *Tracking vs. averaging.* There is also a tradeoff between "tracking" and "averaging". Some parts of the environment change so rapidly relative to a given subsystem's response rate that the sub-system can only react to the average effect; in other situations the subsystem can actually change fast enough to respond "move by move". Again the relative proportion of these two possibilities in the niches the subsystem inhabits has much to do with the subsystem's life history.

(7) *Nonlinearity.* The value ("fitness") of a given combination of building blocks often cannot be predicted by a summing up of values assigned to the component blocks. This nonlinearity (commonly called *epistasis* in genetics) leads to co-adapted sets of blocks (*alleles*) that serve to bias sampling and add additional layers to the hierarchy.

(8) *Coupling.* At all levels, the competitive interactions give rise to counterparts of the familiar interactions of population biology -- *symbiosis*, *parasitism*, *competitive exclusion*, and the like.

(9) *Generalists and specialists.* Subsystems can often be usefully divided into *generalists* (averaging over a wide variety of situations, with a consequent high sampling rate and high statistical confidence, at the cost of a relatively high error rate in individual situations) and *specialists* (reacting to a restricted class of situations with a lowered error rate, bought at the cost of a low sampling rate).

(10) *Multifunctionality.* Subsystems often exhibit multifunctionality in the sense that a given combination of building blocks can usefully exploit quite distinct niches (environmental regularities), typically with different efficiencies. Subsequent recombinations can produce specializations that emphasize one function, usually at the cost of the other. Extensive changes in behavior and efficiency, together with extensive *adaptive radiation,* can result from recombinations involving these multifunctional founders.

(11) *Internal models.* ANNs usually generate implicit internal models of their environments, models progressively revised and improved as the system accumulates experience. The systems *learn.* Consider the progressive improvements of the immune system when faced with antigens, and the fact that one can infer much about the system's environment and history by looking at the antigen population. This ability to infer something of a system's environment and history from its changing internal organization is the diagnostic feature of an implicit internal model. The models encountered are usually *prescriptive* -- they specify preferred responses to given environmental states -- but, for more complex systems (the CNS, for example), they may also be more broadly *predictive,* specifying the results of alternative courses of action. The relevant mathematical concept of a model of process-like transformations is that of a *homomorphism.* Real systems almost never admit of models meeting the requirements for a homomorphism ("commutativity of the diagram"), but there are weakenings, the so-called *q-morphisms* (*quasi-homomorphisms*). The origin of a hierarchy can be looked upon as a sequence of progressively refined q-morphisms (specifically q-morphisms of Markov processes) based upon observation.

Functional Extensions.

The foregoing questions and commonalities, together with some of the problems already encountered in simulations, have already suggested extensions of the standard definitions (as in Holland [1980]) of classifiers systems.

One important change involves the way bids are used in determining the winners of competititions for activation. The standard way of doing this is to calculate a *bid* = [bid ratio]*[strength]. Under this arrangement, the local fixed points of classifiers are such that a generalist and a specialist active in the same situations will come to bid the same amount (because the strength of the generalist increases to the point of compensating for its smaller bid ratio, see the Appendix). This goes against the dictum that specialists should be favored in a competition with generalists. To compensate for this an *effective bid* is calculated by reducing the *bid* in proportion to the generality of the classifier producing the bid. The *effective bid* is then used in determining the probability that the classifier generating it is one of the winners of the competition. If the classifier wins it must pay the bid, *not* the effective bid, to its suppliers under the bucket brigade. Thus, the local fixed points are not changed, but specialists *are* favored in competition with generalists. This change goes a long way toward reducing instabilities in emergent default hierarchies. (We are still exploring the effects in simulations and, at the level of theory, the resulting modifications in global fixed points).

A related change concerns the method of determining a classifier's probability of producing offspring, its fitness, under the genetic algorithm. The higher strength of a generalist at its local fixed point greatly favors it in the production of offspring, and simulations indicate that this overbiases the evolution of the system toward the offspring of generalists. The simplest way of compensating for this is to make the fitness proportional to [bid ratio]*[strength] rather than strength alone. In intuitive terms, this makes the fitness proportional to the classifier's potential for affecting the system (its bid can be thought of as a "phenotypic" effect), rather than its reserves (strength is a quantity determined by its "genotypic" fixed-point). We have yet to carry out an organized set of simulations based on fitness so-determined.

Simulations have also revealed two other effects worth systematic investigation. The first of these is the "focussing" effect of the size of the message list (see R. Riolo's paper in this Proceedings). In effect, a small message list forces the system to concentrate on a few factors in the current situation. Clearly there is the possibility of making the size of the message list depend upon the "urgency" of the situation. For example, during "lookahead" the message list's size can be quite large to encourage an exploration of possibilities, while at "execution" time the size can be reduced to enforce a decision. Clearly, the system can use classifiers to control the size of the message list. This makes the size dependent upon the system's "reading" of the current situation, and the "reading" is subject to long-term adaptive change under the genetic algorithm.

A second simple effect is to revise the definition of the environment, or equivalently the definition of the system's speed, so that typical stimuli persist for several time-steps. (This corresponds to the fact that the CNS operates rapidly relative to typical changes in its environment -- usually, milliseconds vs. tenths of a second.) The resulting "persistence" and "overlap" of input messages makes it much easier for the classifier system to develop causal models and associative links (see below). As yet, to my knowledge, no simulations have been built along these lines.

At a much more general (and speculative) level, use of *triggered* genetic operators provides a major extension of genetic algorithms. Triggering amounts to invoking genetic operators with selected arguments, when certain pre-defined conditions are satisfied.

As an example of a triggering condition consider the following: "Only general classifiers that produce weak bids are activated by the current input message." When this condition occurs it is a sign that the system has little specific information for dealing with the current environmental situation. Let this condition trigger a cross between the input message and the condition parts of some of the active general rules. The result will be plausible new rules with more specific conditions. This amounts to a bottom-up procedure for producing candidate rules that will automatically be tested for usefulness when similar situations recur.

As another example of a triggering condition consider: "Rule C has just made a large profit under the bucket brigade." Satisfaction of this condition signals a propitious time to couple the profitable classifier to its stage-setting precursor. An appropriate cross between the message part of a rule C_0 active on the immediately preceding time-step -- the precursor -- and the condition part of the profit-making successor can produce a new pair of *coupled* rules. (The trigger is *not* activated if C_0 is already coupled to C). The coupled, offspring pair models the state transition mediated by the original pair of (uncoupled) rules. Such coupled rules can serve as the building blocks for models of the environment. Because the couplings serve as "bridges" for the bucket brigade, these building blocks will be assigned credit in accord with the efficacy of the models constructed from them. Interestingly enough there seems to be a rather small number of robust triggering conditions (see Holland et al. [1986]), but each of them would appear to add substantially to the responsiveness of the classifier system.

Tags are particularly affected by triggering conditions that provide new couplings. Tags serve as the glue of larger systems, providing both associative and temporal (model-building) pointers. Under certain kinds of triggered coupling the message sent by the precursor in the coupled pair can have a "hash-coded" section (say a prefix or suffix). The purpose of this hash-coded tag is to prevent accidental eavesdropping by other classifiers -- a sufficient number of randomly generated bits in the tag will prevent accidental matches with other conditions (unless the tag region in the condition part of the potential eavesdropper consists mostly #'s). If the coupled pair proves useful to the system then it will have further offspring under the genetic algorithm, and these offspring often will be coupled to other rules in the system. Typically, the tag will be passed on to the offspring, serving as a common element in all the couplings. The tag will only persist if the resulting cluster of rules proves to be a useful "subroutine". In this case, the "subroutine" can be "called" by messages that incorporate the tag, because the conditions of the rules in the cluster are satisfied by such messages.

In short, the tag that was initially determined at random now "names" the developing subroutine. It even has a *meaning* in terms of the actions it calls forth. Moreover, the tag is subject to the same kinds of recombination as other parts of the rules (it is, after all, a schema). As such it can serve as a building block for other tags. It is as if the system were inventing symbols for its internal use. Clearly, any simulation that provides for a test of these ideas will be an order of magnitude more sophisticated than anything we have tried to date. Runs involving hundreds of thousands of time-steps and thousands of classifiers will probably be required to test these ideas.

Support is another technique that adds considerably to the system's flexibility. Basically, support is a technique that enables the classifier system to integrate many pieces of partial information (such as several views of a partially obscured object) to arrive at strong conclusions. Support is a quantity that travels *with* messages, rather than being a counterflow as in the case of bids. When a classifier is satisfied by several messages from the message list, each such message adds its support into that classifier's *support counter*. Unlike a classifier's strength, the support accrued by a classifier lasts for only the time-step in which it is accumulated. That is, the support counter is reset at the end of each time-step (other techniques are possible, such as a long or short half-life). Support is used to modify the size of the classifier's bid on that time-step; large support increases the bid, small support decreases it. If the classifier wins the bidding competition, the message it posts carries a support proportional to the size of its bid. The propagation of support over sets of coupled classifiers acts somewhat like spreading activation (see Anderson [1983]), but it is much more directed. It can bring associations (coupled rules) into play while serving its primary mission of integrating partial information (messages from several weakly-bidding, general rules that satisfy the same classifier).

In addition to these broadly conceived extensions, there are more special extensions that may have global consequences, particularly in respect to increased responsiveness and robustness.

One of these concerns a simple redefinition of classifiers. The standard definition of a 2-condition classifier requires that *each* condition be satisfied by some message on the message list, in effect an AND, requiring a message of type X and a message of type Y. It is a simple thing to replace the implicit AND with other string operators, e.g. a bit-by-bit AND or a binary sum of the satisfying messages, which is then passed through as the outgoing message. This extension has been implemented, but has not been systematically tested.

Other simple extensions impact the functioning of the genetic algorithm. It is easy to introduce, in the string defining a classifier, *punctuation marks* that bias the probability of crossover (say crossover is twice as likely to take place adjacent to a punctuation mark). These punctuation marks are not interpreted in executing the classifier, but they bias the form of its offspring under the genetic algorithm. Punctuation marks can be treated as alleles under the genetic algorithm, subject to mutation, crossover, etc., just as the other (function-defining) alleles. This ensures that the placement of punctuation marks is adaptively determined. Similarly one can introduce *mating tags* that restrict crossover to classifiers with similar tags; again the tags, as part of the classifier, can be made subject to modification and selection by the genetic algorithm.

Finally, there are two broad ranges of investigation, far beyond anything we yet understand either theoretically or empirically, that offer intriguing possibilities for the future. One of these stems from the fact that classifier systems are general-purpose. They can be programmed initially to implement whatever expert knowledge is available to the designer; learning then allows the system to expand, correct errors, and transfer information from one domain to another. It is important to provide ways of instructing such systems so that they can generate rules -- tentative hypotheses -- on the basis of advice. It is also important that we understand how lookahead and virtual explorations can be incorporated without disturbing other activities of the system. Little has been done in either direction.

The other realm of investigation concerns fully-directed rule generation. In a precursor of classifier systems, the *broadcast language* (Holland [1975]), provision was made for the generation of rules by other rules. With minor changes to the definition of classifier systems, this possibility can be reintroduced. (Both messages and rules are strings. By enlarging the message alphabet, lengthening the message string, and introducing a special symbol that indicates whether a string is to be interpreted as a rule or a message, the task can be accomplished.) With this provision the system can invent its own candidate operators and rules of inference. Survival of these meta- (operator-like) rules should then be made to depend on the net usefulness of the rules they generate (much as a schema takes it value from the average value of its carriers). It is probably a matter of a decade or two before we can do anything useful in this area.

Mathematical Extensions.

There are at least two broader mathematical tasks that should be undertaken. One is an attempt to produce a general characterization of systems that exhibit *implicit parallelism*. Up to now all such attempts have led to sets of algorithms that are easily recast as genetic algorithms -- in effect, we still only know of one example of an algorithm that exhibits implicit parallelism.

The second task involves developing a mathematical formulation of the process whereby a system develops a useful internal model of an environment exhibiting perpetual novelty. In our (preliminary) experiments to date, these models typically exhibit a (tangled) hierarchical structure with associative couplings. As mentioned earlier, such structures can be characterized mathematically as quasi-homomorphisms (see Holland et al. [1986]). The perpetual novelty of the environment can be characterized by a Markov process in which each state has a recurrence time that is large relative to any feasible observation time. Considerable progress can be made along these lines (see Holland [1986b]), but much remains to be done. In particular, we a need to construct an interlocking set of theorems based on: (1) a more global set of fixed point theorems that relates the

strengths of classifiers under the bucket brigade to observed payoff statistics;
(2) a set of theorems that relates building blocks exploited by the "slow" dynamics of the genetic algorithm to the sampling rates for rules at different levels of the emerging default hierarchy (more general rules are tested more often); and
(3) a set of theorems (based on the previous two sets) that detail the way in which various kinds of environmental regularities are exploited by the genetic algorithm acting in terms of the strengths assigned by the bucket brigade.

Appendix.

A simplified version of the fundamental theorem for genetic algorithms can be stated as follows (for an explanation of terms, see Holland [1975] or Holland [1986a]).

Theorem (*Implicit parallelism*). Given a *fitness function* $u: \{0,1\}^k \to Reals^+$, a *population* $B(t)$ of M strings drawn from the set $\{0,1\}^k$, and any *schema* $s \in \{0,1,*\}^k$ defining a hyperplane in $\{0,1\}^k$,

$$\mathbf{M_s(t+1) \geq u^*_s(t)\epsilon_s M_s(t)},$$

where $M_s(t+1)$ is the expected number of instances of s in $B(t+1)$,

$$u^*_s(t) = \sum_{b \in s \text{ AND } b \in B(t)} u(b)/M_s(t),$$

is the average observed fitness of the instances of schema s in $B(t)$, and

$$\epsilon_s = (1-(k_s-1)P_{cross})/(k-1)),$$

is a "copying error" induced by crossover, where P_{cross} is a constant of the genetic algorithm (often $P_{cross} = 1$) giving the proportion of strings undergoing crossover in a given generation, and $k_s - 1$ is the number of crossover points between the outermost defining symbols of $s \in \{0,1,*\}^k$.

Under interpretation, the implicit parallelism theorem says that the sampling rate for *every* schema with instances in the population is expected to increase or decrease at a rate specified by its observed average fitness, with an error proportional to its defining length.

Theorem (*Speedup*). The number of schemas processed with an error $< \epsilon$ under a genetic algorithm considerably exceeds M^3 for a population of size $M = 2^{\frac{1}{2}k'}$, where $\epsilon = k'/k$.

Theorem (*Bucket brigade local fixed-point*). If, under the bucket brigade algorithm, I_C is the long-term average income (after taxes) of a classifier C, and r_C is its bid-ratio, then its strength S_C will approach I_C/r_C.

Theorem (*q-morphism parsimony*; for definitions, see Holland et al. [1986]). A q-morphism of n levels, in which each successive level uses k or fewer additional variables to define exceptions to the previous level, and in which the rules at each level are correct over at least a proportion p of the instances satisfying them, requires no more than

$\sum_j^n 2^{jk}(1-p)^{j-1}$ rules. (A homomorphism defined on nk variables requires 2^{nk} rules). For n=10, k=2, p=0.5, the q-morphism requires fewer than 2^{12} rules, while a corresponding homomorphism would require 2^{20} rules; that is, the homomorphism would require at least 256 times as many rules as the q-morphism.

References.

Anderson, J. R. [1983]. *The Architecture of Cognition*. Cambridge, Massachusetts: Harvard University Press.

Holland, J. H. [1975]. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Holland, J. H. [1980]. "Adaptive algorithms for discovering and using general patterns in growing knowledge-bases." *International J. Policy Analysis and Information Systems*, *4*, 2. 217-240.

Holland, J. H. [1986a]. "Escaping brittleness: The possibilities of general purpose algorithms applied to parallel rule-based systems." Ch. 20 in *Machine Learning II* (eds. Michalski, R. S., et al.) Los Altos: Morgan Kaufmann.

Holland, J. H. [1986b]. "A mathematical framework for studying learning in classifier systems." *Physica 22D* 307-317.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., and Thagard, P. R. [1986]. *Induction: Processes of Inference, Learning, and Discovery*. Cambridge, Massachusetts: MIT Press.

Acknowledgments.

Much of the research reported here has been supported by the National Science Foundation, currently under grant IRI 8610225. The author also wants to thank the Los Alamos National Laboratory for a year (1986-87) as Ulam Scholar at the Center for Nonlinear Studies, a year to pursue his chosen research objectives without hinderances.